Transfert d'Informations en Apprentissage de Métriques : une Analyse Théorique

Michaël Perrot[1] et Amaury Habrard[1]

[1]Université Jean Monnet, Laboratoire Hubert Curien UMR CNRS 5516

Résumé

Nous considérons le problème du transfert de connaissances à priori dans le contexte de l'apprentissage supervisé de métriques. Si ce cadre a déjà été appliqué avec succès de manière empirique, il n'existe pas de cadre théorique justifiant une telle approche. Dans ce papier nous proposons une justification théorique basée sur la notion de stabilité algorithmique adaptée pour l'apprentissage supervisé de métriques. Nous présentons une nouvelle définition de la stabilité, on-average-replace-two-stability, qui nous permet de montrer des bornes en généralisation avec un taux de convergence rapide lorsque qu'une métrique source auxiliaire est utilisée pour biaiser le terme de régularisation. De plus, nous dérivons des bornes de consistance qui nous permettent de montrer l'intérêt de considérer une régularisation biaisée pondérée pour laquelle nous présentons une solution pour estimer le poids de la métrique source. Nous vérifions empiriquement l'intérêt de notre approche dans un cadre d'apprentissage de métrique standard et sur un problème d'apprentissage par transfert lorsque seulement quelques étiquettes cibles sont disponibles.

Mots-clef : Apprentissage de métriques, Théorie de l'apprentissage, Stabilité algorithmique, Apprentissage par transfert.

1 Introduction

A lot of machine learning problems, such as clustering, classification or ranking, require to accurately compare examples by means of distances or similarities. Designing a good metric for a task at hand is thus of crucial importance. Manually tuning a metric is in general difficult and tedious, a recent trend consists to learn the metrics directly from data. This has led to the emergence of *supervised metric learning*, see [BHS13, Kul13] for up-to-date surveys. The underlying idea is to infer automatically the parameters of a metric in order to capture the idiosyncrasies of the data. In a supervised classification perspective, this is generally done by trying to satisfy pair-based constraints aiming at assigning a small (resp. large) score to pairs of examples of the same class (resp. different class). Most of the existing work has notably focused on learning Mahalanobis-like distances of the form $d_{\mathbf{M}}(\mathbf{x}, \mathbf{x}') = \sqrt{(\mathbf{x} - \mathbf{x}')^T \mathbf{M} (\mathbf{x} - \mathbf{x}')}$ where $\mathbf{M}$ is a positive semi-definite (PSD) matrix [1], the learned matrix being typically plugged in a k-Nearest Neighbor classifier allowing one to achieve a better accuracy than the standard Euclidean distance.

Recently, there is a growing interest for methods able to take into account some background knowledge [PW10, CYL13, BYGP14] for learning $\mathbf{M}$. This is in particular the case for *supervised regularized metric learning approaches* where the regularizer is biased with respect to an auxiliary metric given under the form of a matrix. The main objective here is to make use of this a priori knowledge in a setting where *only few labelled data* are available to help learning. For example, in the context of learning a PSD matrix $\mathbf{M}$ plugged into a Mahalanobis-like distance as discussed above, let $\mathbf{I}$ be the identity matrix used as an auxiliary knowledge, $\|\mathbf{M} - \mathbf{I}\|$ is a biased regularizer often considered. This regularization can be interpreted as follows : learn $\mathbf{M}$ while trying to stay close to the Euclidean distance, or from another standpoint try to learn a matrix $\mathbf{M}$ which performs better than $\mathbf{I}$. Other standard matrices can be used such as Σ^{-1} the inverse of the variance-covariance matrix, note that if we take the $\mathbf{0}$ matrix, we retrieve the classical unbiased regularization term.

Another useful setting comes when $\mathbf{I}$ is replaced by any auxiliary matrix $\mathbf{M}_{\mathcal{S}}$ learned from another task.

1. Note that this distance is a generalization of some well-known distances : when $\mathbf{M} = \mathbf{I}$, $\mathbf{I}$ being the identity matrix, we retrieve the Euclidean distance, when $\mathbf{M} = \Sigma^{-1}$ where Σ is the variance-covariance matrix of the data at hand, it actually corresponds to the original definition of a Mahalanobis distance.

This corresponds to a *transfer learning* approach where the biased regularization can be interpreted as transferring the knowledge brought by $\mathbf{M}_\mathcal{S}$ for learning $\mathbf{M}$. This setting is appropriate when the distributions over training and testing domains are different but related. *Domain adaptation* strategies [BBC+10] propose to make use of the relationship between the training examples, called the *source domain*, and the testing instances, called the *target domain* to infer a model. However, it is sometimes not possible to have access to all the training examples, for example when some new domains are acquired incrementally. In this context, transferring the information directly from the model learned from the source domain without any other access to the source domain is of crucial importance. In the context of this paper, we call this setting *Metric Hypothesis Transfer Learning* in reference to the *Hypothesis Transfer Learning* model introduced in [KO13] in the context of classical supervised learning.

Metric learning generally suffers from a lack of theoretical justifications, in particular *metric hypothesis transfer learning* has never been investigated from a theoretical standpoint. In this paper, we propose to bridge this gap by providing a theoretical analysis showing that *supervised regularized metric learning* approaches using a biased regularization are well-founded. Our theoretical analysis is based on *algorithmic stability* arguments allowing one to derive generalization guarantees when a learning algorithm does not suffer too much from a little change in the training sample. As a first contribution, we introduce a new notion of stability called *on-average-replace-two-stability* that is well-suited to regularized metric learning formulations. This notion allows us to prove a high probability generalization bound for metric hypothesis transfer learning achieving a fast converge rate in $\mathcal{O}(1/n)$ in the context of admissible, lipschitz and convex losses. In a second step, we provide a consistency result from which we justify the interest of *weighted biased regularization* of the form $\|\mathbf{M} - \beta\mathbf{M}_\mathcal{S}\|$ where β is a parameter to set. From this result, we derive an approach for assessing this parameter without resorting to a costly parameter tuning procedure. We also provide an experimental study showing the effectiveness of transfer metric learning with weighted biased regularization in the presence of few labeled data both on standard metric learning and transfer learning tasks.

This paper is organized as follows. Section 2 introduces some notations and definitions while Section 3 discusses some related work. Our theoretical analysis is presented in Section 4. We detail our experiments in Section 5 before concluding in Section 6.

2 Notations and Definitions

We start by introducing several notations and definitions that will be used throughout the paper. Let $\mathcal{T}$ be a domain equipped with a probability distribution $\mathcal{D}_\mathcal{T}$ defined over $\mathcal{X} \times \mathcal{Y}$, where $\mathcal{X} \subseteq \mathbb{R}^d$ and $\mathcal{Y}$ is the label set. We consider metrics corresponding to distance functions $\mathcal{X} \times \mathcal{X} \to \mathbb{R}^+$ parameterized by a $d \times d$ positive semi-definite (PSD) matrix $\mathbf{M}$ denoted $\mathbf{M} \succeq 0$. In the following, a metric will be represented by its matrix $\mathbf{M}$. We also consider that we have access to some additional information under the form of an auxiliary $d \times d$ matrix $\mathbf{M}_\mathcal{S}$, throughout this paper we call this additional information source metric or source hypothesis. We denote the Frobenius norm by $\|\cdot\|_\mathcal{F}$, $\mathbf{M}_{kl}$ represents the value of the entry at index (k, l) in matrix $\mathbf{M}$, $[a]_+ = max(a, 0)$ denotes the hinge loss and $[n]$ the set $\{1, \ldots, n\}$ for any $n \in \mathbb{N}$.

Let $T = \{\mathbf{z}_i = (\mathbf{x}_i, y_i)\}_{i=1}^n$ be a labeled training set drawn from $\mathcal{D}_\mathcal{T}$. We consider the following learning framework for *biased regularized metric learning* :

$$\mathbf{M}^* = \underset{\mathbf{M} \succeq 0}{\arg\min}\, L_T(\mathbf{M}) + \lambda \|\mathbf{M} - \mathbf{M}_\mathcal{S}\|_\mathcal{F} \qquad (1)$$

where $L_T(\mathbf{M}) = \frac{1}{n^2} \sum_{\mathbf{z},\mathbf{z}' \in T} l(\mathbf{M}, \mathbf{z}, \mathbf{z}')$ stands for the empirical risk of a metric hypothesis $\mathbf{M}$. Similarly we denote the true risk by $L_{\mathcal{D}_\mathcal{T}}(\mathbf{M}) = \mathbb{E}_{\mathbf{z},\mathbf{z}' \sim \mathcal{D}_\mathcal{T}} l(\mathbf{M}, \mathbf{z}, \mathbf{z}')$. In this work we only consider convex, k-lipschitz and (σ, m)-admissible losses for which we recall the definitions below.

Definition 1 (k-lipschitz continuity)**.** *A loss function $l(\mathbf{M}, \mathbf{z}, \mathbf{z}')$ is k-lipschitz w.r.t. its first argument if, for any matrices $\mathbf{M}$, $\mathbf{M}'$ and any pair of examples $\mathbf{z}, \mathbf{z}'$, there exists $k \geq 0$ such that :*

$$|l(\mathbf{M}, \mathbf{z}, \mathbf{z}') - l(\mathbf{M}', \mathbf{z}, \mathbf{z}')| \leq k\|\mathbf{M} - \mathbf{M}'\|_\mathcal{F}.$$

This property ensures that the loss deviation does not exceed the deviation between matrices $\mathbf{M}$ and $\mathbf{M}'$ with respect to a positive constant k.

Definition 2 ((σ, m)-admissibility)**.** *A loss function $l(\mathbf{M}, \mathbf{z}, \mathbf{z}')$ is (σ, m)-admissible, w.r.t. $\mathbf{M}$, if it is convex w.r.t. its first argument and if for any two pairs of examples $\mathbf{z}_1, \mathbf{z}_2$ and $\mathbf{z}_3, \mathbf{z}_4$, we have :*

$$|l(\mathbf{M}, \mathbf{z}_1, \mathbf{z}_2) - l(\mathbf{M}, \mathbf{z}_3, \mathbf{z}_4)| \leq \sigma\,|y_1 y_2 - y_3 y_4| + m$$

where $y_i y_j = 1$ if $y_i = y_j$ and -1 otherwise. Thus $|y_1 y_2 - y_3 y_4| \in \{0, 2\}$.

This property bounds the difference between the losses of two pairs of examples by a value only related to the labels plus a constant independent from $\mathbf{M}$.

To derive our theoretical results, we make use of the notion of *algorithmic stability* which allows one to provide generalization guarantees. A learning algorithm is stable if a slight modification in its input does not change its output much. In our analysis we use two definitions of stability. On the one hand, we introduce in Section 4.1 the notion of *on-average-replace-two-stability* which is an adaptation to metric learning of the notion of on-average-replace-one-stability proposed in [SSBD14] and recalled in Def. 3 below.

Definition 3 (On-average-replace-one-stability)**.** *Let $\epsilon : \mathbb{N} \to \mathbb{R}$ be monotonically decreasing and $U(n)$ be the uniform distribution over $[n]$. An algorithm A is on-average-replace-one-stable with rate $\epsilon(n)$ if for any distribution $\mathcal{D}_\mathcal{T}$*

$$\mathbb{E}_{\substack{T\sim\mathcal{D}_\mathcal{T}^n \\ i\sim U(n) \\ \mathbf{z}'\sim\mathcal{D}_\mathcal{T}}} \left[l(A(T^i), \mathbf{z}^i) - l(A(T), \mathbf{z}^i) \right] \leq \epsilon(n)$$

where $A(T)$, respectively $A(T^i)$ is the optimal solution of algorithm A when learning with training set T, respectively T^i. T^i is obtained by replacing the i^{th} example of T by $\mathbf{z}'$.

This property ensures that, given an example, learning with or without it will not imply a big change in the hypothesis prediction. Note that the property is required to be true on average over all the possible training sets of size n.

On the other hand, we consider an adaptation of the framework of *uniform stability* for metric learning proposed in [JWZ09] and recalled in Def. 4.

Definition 4 (Uniform stability)**.** *A learning algorithm has a uniform stability in $\frac{\mathcal{K}}{n}$, with $\mathcal{K} \geq 0$ a constant, if $\forall i$,*

$$\sup_{\mathbf{z},\mathbf{z}'\sim\mathcal{D}_\mathcal{T}} \left| l(\mathbf{M}^*, \mathbf{z}, \mathbf{z}') - l(\mathbf{M}^{i*}, \mathbf{z}, \mathbf{z}') \right| \leq \frac{\mathcal{K}}{n}$$

where $\mathbf{M}^*$ is the matrix learned on the training set T, $\mathbf{M}^{i*}$ is the matrix learned on the training set T^i obtained by replacing the i^{th} example of T by a new independent one.

Uniform stability requires that a small change in the training set does not imply a significant variation in the learned models output. The constraint in $\mathcal{O}\left(\frac{1}{n}\right)$ over the supremum makes this property rather strong since it considers a worst case over the possible pairs of examples to compare, whatever the training set. It is actually one of the most general algorithmic stability setting [BE02].

3 Related Work

3.1 Metric Learning

Based on the pioneering approach of [XNJR02], metric learning aims at finding the parameters of a distance function by maximizing the distance between dissimilar examples (*i.e.* examples of different class) while maintaining a small distance between similar ones (*i.e.* of similar class). Following this idea, one of the most famous approach, called LMNN [WBS05], proposes to learn a PSD matrix dedicated to improve the k-nearest neighbours algorithm. To do so, the authors force the metric to respect triplet-based local constraints of the form $(\mathbf{z}_i, \mathbf{z}_j, \mathbf{z}_k)$ where $\mathbf{z}_j$ and $\mathbf{z}_k$ belong to the neighbourhood of $\mathbf{z}_i$, $\mathbf{z}_i$ and $\mathbf{z}_j$ being of the same class, and $\mathbf{z}_k$ being of opposite class. The constraints impose that $\mathbf{z}_i$ should be closer to $\mathbf{z}_j$ than to $\mathbf{z}_k$ with respect to a margin ε. In ITML, [DKJ$^+$07] propose to use a Log-Det divergence as a regularizer allowing one to ensure an automatic enforcement of the PSD constraint. The idea is to force the learned matrix $\mathbf{M}$ to stay as close as possible to a good matrix $\mathbf{M}_\mathcal{S}$ defined a-priori (in general $\mathbf{M}_\mathcal{S}$ is chosen as the identity matrix). Indeed, if this divergence is finite, the authors show that $\mathbf{M}$ is guaranteed to be PSD. This constraint over $\mathbf{M}$ can be interpreted as a biased regularization w.r.t. $\mathbf{M}_\mathcal{S}$.

The idea behind biased regularization has been successfully used in many metric learning approaches. For example, [ZMW$^+$09] have proposed to replace the identity matrix ($\mathbf{M}_\mathcal{S} = \mathbf{I}$) originally used in ITML by matrices previously learned on so called auxiliary data sets. Similarly, in [PW10] the authors are interested in Multi-Task metric learning. They propose to learn one metric for each task and a global metric common to all the tasks. For this global metric, they consider a biased regularization of the form $\|\mathbf{M} - \mathbf{I}\|_\mathcal{F}^2$ where $\mathbf{I}$ is the identity matrix but they do not study any other kind of source information. In [CYL13], the authors use a similar biased regularization to learn a metric learning model for face recognition. As a last example, [BYGP14] introduce a regularization of the form $\|\mathbf{M} - \beta\mathbf{I}\|_\mathcal{F}$ where they learn $\mathbf{M}$ and β. In our work, instead of optimizing these two parameters, we derive a theoretically founded algorithm to choose beforehand the optimal value of β.

3.2 Theoretical Frameworks in Metric Learning

Theoretically speaking, there is not a lot of frameworks for metric learning. The goal of generalization guarantees is to show that the empirical estimation of

the error of an algorithm does not deviate much from the true error. One of the main difficulty in deriving bounds for metric learning is the fact that instead of considering examples drawn i.i.d. from a distribution, we consider pairs of examples which might not be independent. Building upon the framework of stability proposed in [BE02], [JWZ09] propose one of the first study of the generalization ability of a metric learning algorithm. Building upon this work, [PHMS14] give theoretical guarantees for a local metric learning algorithm and [BHS12] derive generalization guarantees for a similarity learning algorithm. Other ways to derive generalization guarantees are to use the Rademacher complexity as in [CGY12, GY14] or to use the notion of algorithmic robustness [BH15].

3.3 Biased Regularization in Supervised Learning

Biased regularization has already been studied in non metric learning settings. For example in [KC06], the authors propose to use biased regularization to learn SVM classifiers. A first theoretical study of biased regularization in the context of regularized least squares has been proposed in [KO13]. Their study is based on a notion of *hypothesis stability* less general than the *uniform stability* used in our approach. In [KO14], the authors derive generalization bounds based on the Rademacher complexity for regularized empirical risk minimization methods in a supervised learning setting. Their results show that if the true risk of the source hypothesis on the target domain is low, then the generalization rate can be improved. However computing the true risk of the source hypothesis is not possible in practice. In our analysis, we derive a generalization bound which depends on the empirical risk and the complexity (w.r.t. the regularization term) of the source metric. It allows us to derive an algorithm to minimize the generalization bound taking into account the performance and the complexity of the source metric.

4 Contribution

We divide our contribution consisting of a theoretical analysis of Alg. 1 given convex, k-lipschitz and (σ, m)-admissible losses into three parts. First, we provide an on average analysis for $\mathbb{E}_T\left[L_{\mathcal{D}_\mathcal{T}}(\mathbf{M}^*)\right]$ where $\mathbf{M}^*$ represents the metric learned with Alg. 1 using training set T. This analysis allows us to bound the expected loss over distribution $\mathcal{D}_\mathcal{T}$ with respect to the loss of the auxiliary metric $\mathbf{M}_\mathcal{S}$ over $\mathcal{D}_\mathcal{T}$. It shows that on average the learned metric tends to be better than the given source $\mathbf{M}_\mathcal{S}$, with a fast convergence rate in $\mathcal{O}(1/n)$. Second, we provide a consistency analysis of our framework leading to a standard convergence rate of $\mathcal{O}\left(\frac{1}{\sqrt{n}}\right)$ w.r.t the empirical loss over T optimized in Alg. 1. In a third part, we specialize the previous consistency result to a specific loss and show that it is possible to refine our generalization bound in order to depend both on the complexity of our source metric $\mathbf{M}_\mathcal{S}$ and its empirical performance on the training set T. We then deduce an approach to weight the importance of the source hypothesis for optimizing the generalization bound.

4.1 On average analysis

Def. 3 allows one to perform an average analysis over the expected loss, however its formulation is not tailored to metric learning approaches that work with pair of examples. Thus we propose an adaptation of it that we call *on-average-replace-two-stability* allowing one to derive sharp bounds for metric learning.

Definition 5 (On-average-replace-two-stability)**.** *Let $\epsilon : \mathbb{N} \to \mathbb{R}$ be monotonically decreasing and let $U(n)$ be the uniform distribution over $[n]$. A metric learning algorithm is on-average-replace-two-stable with rate $\epsilon(n)$ if for every distribution $\mathcal{D}_\mathcal{T}$*

$$\mathbb{E}_{\substack{T\sim\mathcal{D}_\mathcal{T}{}^n \\ i,j\sim U(n) \\ \mathbf{z}_1,\mathbf{z}_2\sim\mathcal{D}_\mathcal{T}}}\left[l(\mathbf{M}^{ij^*}, \mathbf{z}^i, \mathbf{z}^j) - l(\mathbf{M}^*, \mathbf{z}^i, \mathbf{z}^j)\right] \leq \epsilon(n)$$

where $\mathbf{M}^*$, respectively $\mathbf{M}^{ij^*}$, is the optimal solution when learning with the training set T, respectively T^{ij}. T^{ij} is obtained by replacing $\mathbf{z}^i$, the i^{th} example of T, by $\mathbf{z}_1$ to get a training set T^i and then by replacing $\mathbf{z}^j$, the j^{th} example of T^i, by $\mathbf{z}_2$.

Note that when this definition is true, it implies $\mathbb{E}_T\left[L_{\mathcal{D}_\mathcal{T}}(\mathbf{M}^*) - L_T(\mathbf{M}^*)\right] \leq \epsilon(n)$. The next theorem shows that our algorithm is on-average-replace-two-stable.

Theorem 1 (On-average-replace-two-stability)**.** *Given a training sample T of size n drawn i.i.d. from $\mathcal{D}_\mathcal{T}$, our algorithm is on-average-replace-two-stable with $\epsilon(n) = \frac{8k^2}{\lambda n}$.*

Démonstration. The proof of Th. 1 can be found in the supplementary material. □

We can now bound the expected true risk of our algorithm.

Theorem 2 (On average bound)**.** *For any convex, k-lipschitz loss, we have :*

$$\mathbb{E}_{T\sim\mathcal{D}_\mathcal{T}^n}\left[L_{\mathcal{D}_\mathcal{T}}(\mathbf{M}^*)\right] \leq L_{\mathcal{D}_\mathcal{T}}(\mathbf{M}_\mathcal{S}) + \frac{8k^2}{\lambda n}$$

where the expected value is taken over size-n training sets.

Démonstration. We have :

$$\begin{aligned}
&\mathbb{E}_T\left[L_{\mathcal{D}_\mathcal{T}}(\mathbf{M}^*)\right] \\
&\quad= \mathbb{E}_T\left[L_{\mathcal{D}_\mathcal{T}}(\mathbf{M}^*)\right] + \mathbb{E}_T\left[L_T(\mathbf{M}^*)\right] - \mathbb{E}_T\left[L_T(\mathbf{M}^*)\right] \\
&\quad= \mathbb{E}_T\left[L_T(\mathbf{M}^*)\right] + \mathbb{E}_T\left[L_{\mathcal{D}_\mathcal{T}}(\mathbf{M}^*) - L_T(\mathbf{M}^*)\right] \\
&\quad\leq \mathbb{E}_T\left[L_T(\mathbf{M}_\mathcal{S})\right] + \frac{8k^2}{\lambda n}. \qquad (2)
\end{aligned}$$

Inequality 2 is obtained by noting that from Th. 1 we have $\mathbb{E}_T\left[L_{\mathcal{D}_\mathcal{T}}(\mathbf{M}^*) - L_T(\mathbf{M}^*)\right] \leq \frac{8k^2}{\lambda n}$, then the convexity of our algorithm and the optimality of $\mathbf{M}^*$ give $L_T(\mathbf{M}^*) \leq L_T(\mathbf{M}^*) + \lambda\|\mathbf{M}^* - \mathbf{M}_\mathcal{S}\|_\mathcal{F}^2 \leq L_T(\mathbf{M}_\mathcal{S}) + \lambda\|\mathbf{M}_\mathcal{S} - \mathbf{M}_\mathcal{S}\|_\mathcal{F}^2$. Noting that $\mathbb{E}_T\left[L_T(\mathbf{M}_\mathcal{S})\right] = L_{\mathcal{D}_\mathcal{T}}(\mathbf{M}_\mathcal{S})$ gives the theorem. □

This bound shows that with a sufficient number of examples w.r.t. a fast convergence rate in $\mathcal{O}(1/n)$, we will on average obtain a metric which is at least as good as the source hypothesis. Thus choosing a good source metric is key to learn well.

4.2 Consistency analysis

We now provide a consistency analysis taking into account the empirical risk optimized in Alg. 1. We begin by showing that our algorithm is uniformly stable w.r.t. Def. 4 in the next theorem.

Theorem 3 (Uniform stability)**.** *Given a training sample T of n examples drawn i.i.d. from $\mathcal{D}_\mathcal{T}$, our algorithm has a uniform stability in $\frac{\mathcal{K}}{n}$ with $\mathcal{K} = \frac{4k^2}{\lambda}$.*

Démonstration. The beginning of the proof follows closely the one proposed in [BE02] and is postponed to the supplementary material for the sake of readability. We consider the end of the proof here. We have

$$B \leq \frac{4kt}{n}\|\mathbf{\Delta M}\|_\mathcal{F}$$

where $B = \lambda\|\mathbf{M} - \mathbf{M}_\mathcal{S}\|_\mathcal{F}^2 - \lambda\|\mathbf{M} - t\mathbf{\Delta M} - \mathbf{M}_\mathcal{S}\|_\mathcal{F}^2 + \lambda\|\mathbf{M}^i - \mathbf{M}_\mathcal{S}\|_\mathcal{F}^2 - \lambda\|\mathbf{M}^i + t\mathbf{\Delta M} - \mathbf{M}_\mathcal{S}\|_\mathcal{F}^2$.

Setting $t = \frac{1}{2}$ we have :

$$\begin{aligned}
B &= \lambda\|\mathbf{M} - \mathbf{M}_\mathcal{S}\|_\mathcal{F}^2 - \lambda\|\mathbf{M} - \frac{1}{2}\mathbf{\Delta M} - \mathbf{M}_\mathcal{S}\|_\mathcal{F}^2 \\
&\quad + \lambda\|\mathbf{M}^i - \mathbf{M}_\mathcal{S}\|_\mathcal{F}^2 - \lambda\|\mathbf{M}^i + \frac{1}{2}\mathbf{\Delta M} - \mathbf{M}_\mathcal{S}\|_\mathcal{F}^2 \\
&= \lambda\sum_k\sum_l\Big[(\mathbf{M}_{kl} - \mathbf{M}_{\mathcal{S}kl})^2 - (\mathbf{M}_{kl} - \frac{1}{2}(\mathbf{M}_{kl} - \mathbf{M}^i_{kl}) - \mathbf{M}_{\mathcal{S}kl})^2 \\
&\quad + (\mathbf{M}^i_{kl} - \mathbf{M}_{\mathcal{S}kl})^2 - (\mathbf{M}^i_{kl} + \frac{1}{2}(\mathbf{M}_{kl} - \mathbf{M}^i_{kl}) - \mathbf{M}_{\mathcal{S}kl})^2\Big] \\
&= \lambda\sum_i\sum_j\Big[(\mathbf{M}_{kl} - \mathbf{M}_{\mathcal{S}kl})^2 - (\frac{1}{2}(\mathbf{M}_{kl} - \mathbf{M}_{\mathcal{S}kl}) + \frac{1}{2}(\mathbf{M}^i_{kl} - \mathbf{M}_{\mathcal{S}kl}))^2 \\
&\quad + (\mathbf{M}^i_{kl} - \mathbf{M}_{\mathcal{S}kl})^2 - (\frac{1}{2}(\mathbf{M}_{kl} - \mathbf{M}_{\mathcal{S}kl}) + \frac{1}{2}(\mathbf{M}^i_{kl} - \mathbf{M}_{\mathcal{S}kl}))^2\Big] \\
&= \lambda\sum_i\sum_j\Big[\frac{1}{2}((\mathbf{M}_{kl} - \mathbf{M}_{\mathcal{S}kl})^2 \\
&\quad + (\mathbf{M}^i_{kl} - \mathbf{M}_{\mathcal{S}kl})^2 - 2(\mathbf{M}_{kl} - \mathbf{M}_{\mathcal{S}kl})(\mathbf{M}^i_{kl} - \mathbf{M}_{\mathcal{S}kl}))\Big] \\
&= \lambda\sum_i\sum_j\Big[\frac{1}{2}(\mathbf{M}_{kl} - \mathbf{M}_{\mathcal{S}kl} - \mathbf{M}^i_{kl} - \mathbf{M}_{\mathcal{S}kl})^2\Big] = \frac{\lambda}{2}\|\mathbf{\Delta M}\|_\mathcal{F}^2.
\end{aligned}$$

Then we obtain

$$\frac{\lambda}{2}\|\mathbf{\Delta M}\|_\mathcal{F}^2 \leq \frac{4k}{2n}\|\mathbf{\Delta M}\|_\mathcal{F} \Leftrightarrow \|\mathbf{\Delta M}\|_\mathcal{F} \leq \frac{4k}{\lambda n}.$$

Using the k-lipschitz continuity of the loss, we have :

$$\sup_{\mathbf{z},\mathbf{z}'}|l(\mathbf{M},\mathbf{z},\mathbf{z}') - l(\mathbf{M}^i,\mathbf{z},\mathbf{z}')| \leq k\|\mathbf{\Delta M}\|_\mathcal{F} \leq \frac{4k^2}{\lambda n}.$$

Setting $\mathcal{K} = \frac{4k^2}{\lambda}$ concludes the proof. □

Using the fact that our algorithm is uniformly stable, we can derive generalization guarantees as stated in Th. 4.

Theorem 4 (Generalization bound)**.** *With probability $1-\delta$, for any matrix $\mathbf{M}$ learned with our $\mathcal{K}$ uniformly stable algorithm and for any convex, k-lipschitz and (σ, m)-admissible loss, we have :*

$$L_{\mathcal{D}_\mathcal{T}}(\mathbf{M}) \leq L_T(\mathbf{M}) + (4\sigma + 2m + c)\,\mathcal{O}\left(\frac{1}{\sqrt{n}}\right) + \mathcal{O}\left(\frac{1}{n}\right)$$

where c is a constant linked to the k-lipschitz property of the loss.

Démonstration. The proof of this theorem is available in the supplementary material. □

This bound shows that with a convergence rate in $\mathcal{O}\left(\frac{1}{\sqrt{n}}\right)$ the true risk of our algorithm is bounded above by the empirical risk justifying the consistency of the

approach. In the next section, we propose an extension of this analysis to include the performance of the source metric. This extension allows us to introduce a natural weighting of the source metric in order to improve the proposed bound.

4.3 Refinement with weighted source hypothesis

In this part we study a specific loss, namely $l(\mathbf{M}, \mathbf{z}, \mathbf{z}') = \left[yy'((\mathbf{x}-\mathbf{x}')^T\mathbf{M}(\mathbf{x}-\mathbf{x}') - \gamma_{yy'})\right]_+$ where $yy' = 1$ if $y = y'$ and -1 otherwise. The convexity follows from the use of the hinge loss. In the next two lemmas, we show that this loss is k-lipschitz continuous and (σ, m)-admissible. The (σ, m)-admissibility result is of high importance because it allows us to introduce some information coming from the source matrix $\mathbf{M}_\mathcal{S}$.

Lemma 1 (k-lipschitz continuity)**.** *Let* $\mathbf{M}$ *and* $\mathbf{M}'$ *be two matrices and* $\mathbf{z}, \mathbf{z}'$ *be two examples. Our loss* $l(\mathbf{M}, \mathbf{z}, \mathbf{z}')$ *is* k*-lipschitz continuous with* $k = \max_{\mathbf{x},\mathbf{x}'} \|\mathbf{x}-\mathbf{x}'\|^2$.

Démonstration. The interested reader can find the proof of this lemma in the supplementary material. □

Lemma 2 ((σ, m)-admissibility)**.** *Let* $\mathbf{z}_1, \mathbf{z}_2, \mathbf{z}_3, \mathbf{z}_4$ *be four examples and* $\mathbf{M}^*$ *be the optimal solution of Problem 1. The convex and* k*-lipschitz loss function* $l(\mathbf{M}, \mathbf{z}, \mathbf{z}')$ *is* (σ, m)*-admissible with* $\sigma = \max(\gamma_{y_3y_4}, \gamma_{y_1y_2})$ *and* $m = 2\max_{\mathbf{x},\mathbf{x}'} \|\mathbf{x}-\mathbf{x}'\|^2 \ (\sqrt{\frac{L_T(\mathbf{M}_\mathcal{S})}{\lambda}} + \|\mathbf{M}_\mathcal{S}\|_\mathcal{F})$.

Démonstration. Let $\boldsymbol{\varepsilon}^* = \mathbf{M}^* - \mathbf{M}_\mathcal{S}$ be the difference between the learned metric and the source metric. We first bound the frobenius norm of $\boldsymbol{\varepsilon}^*$ w.r.t. the performance of the source metric.

$$L_T(\mathbf{M}^*) + \lambda\|\mathbf{M}^* - \mathbf{M}_\mathcal{S}\|_\mathcal{F}^2 \leq L_T(\mathbf{M}_\mathcal{S}) + \lambda\|\mathbf{M}_\mathcal{S} - \mathbf{M}_\mathcal{S}\|_\mathcal{F}^2$$

$$\Rightarrow \qquad \lambda\|\boldsymbol{\varepsilon}^*\|_\mathcal{F}^2 \leq L_T(\mathbf{M}_\mathcal{S}) \Leftrightarrow \|\boldsymbol{\varepsilon}^*\|_\mathcal{F} \leq \sqrt{\frac{L_T(\mathbf{M}_\mathcal{S})}{\lambda}}$$

Now we can prove the (σ, m)-admissibility of our loss.

$$\begin{aligned}
&|l(\mathbf{M}^*, \mathbf{z}_1, \mathbf{z}_2) - l(\mathbf{M}^*, \mathbf{z}_3, \mathbf{z}_4)| \\
&= \Big|\left[y_1y_2((\mathbf{x}_1-\mathbf{x}_2)^T\mathbf{M}^*(\mathbf{x}_1-\mathbf{x}_2) - \gamma_{y_1y_2})\right]_+ \\
&\quad - \left[y_3y_4((\mathbf{x}_3-\mathbf{x}_4)^T\mathbf{M}^*(x_3-x_4) - \gamma_{y_3y_4})\right]_+\Big| \\
&\leq |y_1y_2((\mathbf{x}_1-\mathbf{x}_2)^T\mathbf{M}^*(\mathbf{x}_1-\mathbf{x}_2) - \gamma_{y_1y_2}) \\
&\quad - y_3y_4((\mathbf{x}_3-\mathbf{x}_4)^T\mathbf{M}^*(\mathbf{x}_3-\mathbf{x}_4) - \gamma_{y_3y_4})| \qquad (3) \\
&\leq |y_1y_2(\mathbf{x}_1-\mathbf{x}_2)^T\mathbf{M}^*(\mathbf{x}_1-\mathbf{x}_2) \\
&\quad - y_3y_4(\mathbf{x}_3-\mathbf{x}_4)^T\mathbf{M}^*(\mathbf{x}_3-\mathbf{x}_4)| \\
&\quad + |y_3y_4\gamma_{y_3y_4} - y_1y_2\gamma_{y_1y_2}| \\
&\leq 2\max_{\mathbf{x},\mathbf{x}'}((\mathbf{x}-\mathbf{x}')^T\mathbf{M}^*(\mathbf{x}-\mathbf{x}')) \\
&\quad + |y_3y_4 - y_1y_2|\max(\gamma_{y_3y_4}, \gamma_{y_1y_2}) \\
&\leq 2\max_{\mathbf{x},\mathbf{x}'}((\mathbf{x}-\mathbf{x}')^T(\boldsymbol{\varepsilon}^* + \mathbf{M}_\mathcal{S})(\mathbf{x}-\mathbf{x}')) \\
&\quad + |y_3y_4 - y_1y_2|\max(\gamma_{y_3y_4}, \gamma_{y_1y_2}) \\
&\leq 2\max_{\mathbf{x},\mathbf{x}'}\|\mathbf{x}-\mathbf{x}'\|^2(\|\boldsymbol{\varepsilon}^*\|_\mathcal{F} + \|\mathbf{M}_\mathcal{S}\|_\mathcal{F}) \\
&\quad + |y_3y_4 - y_1y_2|\max(\gamma_{y_3y_4}, \gamma_{y_1y_2}) \qquad (4) \\
&\leq 2\max_{\mathbf{x},\mathbf{x}'}\|\mathbf{x}-\mathbf{x}'\|^2(\sqrt{\frac{L_T(\mathbf{M}_\mathcal{S})}{\lambda}} + \|\mathbf{M}_\mathcal{S}\|_\mathcal{F}) \\
&\quad + |y_3y_4 - y_1y_2|\max(\gamma_{y_3y_4}, \gamma_{y_1y_2})
\end{aligned}$$

Inequality 3 comes from the 1-lipschitz property of the hinge loss. We obtain inequality 4 by applying the Cauchy-Schwarz inequality and some classical norm properties. Setting $m = 2\max_{\mathbf{x},\mathbf{x}'} \|\mathbf{x}-\mathbf{x}'\|^2(\sqrt{\frac{L_T(\mathbf{M}_\mathcal{S})}{\lambda}} + \|\mathbf{M}_\mathcal{S}\|_\mathcal{F})$ and $\sigma = \max(\gamma_{y_3y_4}, \gamma_{y_1y_2})$ gives the lemma. □

Using Lemmas 1 and 2 we can now derive, in Th. 5, a generalization bound associated with our specific loss.

Theorem 5 (Generalization bound)**.** *With probability* $1-\delta$ *for any matrix* $\mathbf{M}$ *learned with Alg. 1, we have :*

$$\begin{aligned}
L_{\mathcal{D}_\mathcal{T}}(\mathbf{M}) \leq & L_T(\mathbf{M}) + \mathcal{O}\left(\frac{1}{n}\right) \\
& + \left(\sqrt{\frac{L_T(\mathbf{M}_\mathcal{S})}{\lambda}} + \|\mathbf{M}_\mathcal{S}\|_\mathcal{F} + c_\gamma\right)\mathcal{O}\left(\frac{1}{\sqrt{n}}\right)
\end{aligned}$$

where c_γ *is a constant linked to the* k*-lipschitz property of the loss and the chosen margins.*

Démonstration. The proof is the same as for Th. 4 replacing k, σ and m by their values. □

As for Th. 4, the convergence rate is in $\mathcal{O}\left(\frac{1}{\sqrt{n}}\right)$. The term $C(\mathbf{M}_\mathcal{S}) \stackrel{\text{def}}{=} \left(\sqrt{\frac{L_T(\mathbf{M}_\mathcal{S})}{\lambda}} + \|\mathbf{M}_\mathcal{S}\|_\mathcal{F}\right)$ mainly depends on the quality of the source hypothesis $\mathbf{M}_\mathcal{S}$. The

product $C(\mathbf{M}_\mathcal{S})\mathcal{O}\left(\frac{1}{\sqrt{n}}\right)$ means that as the number of examples available for learning increases, the quality of the source metric is of decreasing importance. A similar result has already been stated in domain adaptation or transfer learning in [BBC+10, KO13] where they show that as the number of target examples increases, the necessity of having source examples decreases.

Given a source hypothesis $\mathbf{M}_\mathcal{S}$, it is possible to optimize it w.r.t. the bound derived in Th. 5. Indeed, note that the term $C(\mathbf{M}_\mathcal{S}) = \left(\sqrt{\frac{L_T(\mathbf{M}_\mathcal{S})}{\lambda}} + \|\mathbf{M}_\mathcal{S}\|_\mathcal{F}\right)$ corresponds to a trade-off between the complexity of the source metric and its performance on the training set. The lower the value of this term, the tighter the bound. Hence, we propose a way to minimize the generalization bound and more specifically $C(\mathbf{M}_\mathcal{S})$ by adding a weighting parameter $\beta \geq 0$ on the source metric $\mathbf{M}_\mathcal{S}$. This parameter is a way to control the trade-off between complexity and performance of the source metric. It can be assessed by means of the following optimization problem :

$$\beta^* = \arg\min_{\beta} C(\beta \mathbf{M}_\mathcal{S}) \tag{5}$$

Note that the bound derived in Th. 5 holds whatever the value of $\mathbf{M}_\mathcal{S}$. Thus replacing it with $\beta^*\mathbf{M}_\mathcal{S}$ does not impact the theoretical study proposed in this section.

Interpretation of the value of β^* We can distinguish three main cases. First if the source hypothesis performs poorly on the training set at hand we expect β^* to be as small as possible to reduce the importance of $\mathbf{M}_\mathcal{S}$. In a sense, we tend to go back to the classical case were $\mathbf{M}_\mathcal{S} = \mathbf{0}$. Second if the source hypothesis is complex and performs well, we expect β^* to be rather small to reduce the complexity of the hypothesis while keeping a good performance on the training set. Third if the source hypothesis is simple and performs well, we expect β^* to be closer to one since $\mathbf{M}_\mathcal{S}$ is already a good choice.

Learning β^* Problem 5 is highly non differentiable [2] and non convex. However, it remains simple in the sense that we have only one parameter to assess and we used a classical subgradient descent to solve it. Even if it is not convex, our empirical study shows no need to perform many restarts to output a good solution : we always found almost the same solution. As a consequence, we applied only one optimization procedure in our experiments.

2. To avoid this problem, we can use the classical relaxation with slack variables.

In this section we presented a new framework for metric learning where one can use a source hypothesis to add some side information during the learning process. We have shown that our approach is consistent with a convergence rate in $\mathcal{O}\left(\frac{1}{\sqrt{n}}\right)$. Furthermore, given a specific loss, we have shown that the use of a weighting parameter to control the importance of the source metric is theoretically founded. In the next part we empirically demonstrate that we can obtain competitive results both in a classical metric learning setting and in a domain adaptation setting.

5 Experiments

We propose an empirical study according to two directions depending on the choice of the source metric. First, using some well-known distances as a source metric, we show that our framework performs well on classical supervised metric learning tasks of the UCI database and we empirically demonstrate the interest of learning the β parameter. Second, we apply our framework in a semi-supervised Domain Adaptation task. We show that, using only source information through a learned metric, our method is able to compete with state of the art algorithms.

Setup In all our experiments we use limited training dataset, making it difficult to apply any kind of cross-validation to set the parameters. Thus we propose to fix them as follows. First the positive and negative margin are respectively set to the 5^{th} and 95^{th} percentile of the training set possible distances computed with the source metric as proposed in [DKJ+07]. Next we set λ such that the two terms of Eq. 5 are equals, i.e. we balance the complexity and performance importance with respect to the source metric. The β parameter is then learned using Algorithm 5. In all the experiments we plug our metric in a 1-nearest neighbour classifier to classify the examples of the test set.

	Breast	Pima	Scale	Wine
# of examples	683	768	625	178
# of classes	2	2	3	3
# of features	9	8	4	13

TABLE 2 – Characteristics of four UCI datasets.

	Baselines			Our approach			
Dataset	1-NN	ITML	LMNN	IDENTITY	IDENTITY-B1	MAHALANOBIS	MAHALANOBIS-B1
Breast	95.31 ± 1.11	95.40 ± 1.37	95.60 ± 0.92	**96.06 ± 0.77**	95.75 ± 0.87	95.71 ± 0.84	94.76 ± 1.38
Pima	67.92 ± 1.95	68.13 ± 1.86	67.90 ± 2.05	67.87 ± 1.57	67.54 ± 1.99	**68.37 ± 2.00**	66.31 ± 2.37
Scale	78.73 ± 1.69	**87.31 ± 2.35**	86.20 ± 2.83	80.98 ± 1.51	80.82 ± 1.27	81.35 ± 1.17	80.88 ± 1.43
Wine	93.40 ± 2.70	93.82 ± 2.63	93.47 ± 1.80	**95.42 ± 1.71**	95.07 ± 1.68	94.31 ± 2.01	80.56 ± 5.75

TABLE 1 – Results of the experiments conducted on the UCI datasets. Each value corresponds to the mean and standard deviation over 10 runs. For each dataset we highlight the best result using a bold font. Approaches with the suffix -B1 do not learn β, it is fixed to 1.

5.1 Classical Supervised Metric Learning

First we start by conducting experiments on several UCI datasets, namely breast, pima, scale and wine. The characteristics of these datasets are reported in Table 2. We propose to consider three source metrics : (i) **Zero** : No source hypothesis, (ii) **Identity** : Euclidean distance, (iii) **Mahalanobis** : Inverse of the variance-covariance matrix computed on the training set.

For the last two hypothesis we propose two experiments, one where we set $\beta = 1$ and one where we learn β using Algorithm 5. The goal of this experiment is to show the interest of automatically setting β. We consider a 1-nearest neighbour (1-NN) classifier using the Euclidean Distance as the baseline and also report the results of two well known metric learning algorithms, namely ITML, [DKJ$^+$07] and LMNN [WBS05]. The results averaged over 10 runs are reported in Table 1. For each run we randomly draw a training set containing 20% of the data available for each class and we test the metric on the remaining 80% of data.

These experiments highlight the interest of learning the β parameter. When we consider the performance of our approach with and without learning β, we mainly notice the following facts. First, learning β always leads to an improvement on all the datasets and the final result is better than the $1NN$ classifier. Second, learning β when considering the identity matrix as the source metric seems to be of limited interest. This can be justified by the fact that, in this case, it only consists of a rescaling of the diagonal of the matrix which does not change much the behaviour of the distance on the dataset. Finally, learning β when considering the variance-covariance matrix as the source metric leads to a significant improvement of the performance of the metric. This is particularly true for the wine dataset with a gain of nearly 14% in accuracy. It can be explained by the fact that, for this dataset, we are learning with less than 40 examples. Thus the original Mahalanobis distance does not carry as much information as in the other datasets and is thus of a lower quality. Learning β allows us to compensate this drawback and to obtain results which are even better than ITML or LMNN.

5.2 Metric learning for Semi-supervised Domain Adaptation

In this section we consider a Semi-supervised Domain Adaptation task with the Office-Caltech dataset. This dataset consists of four domains : Amazon (A), Caltech (C), DSLR (D) and Webcam (W) for which we consider 10 classes. This leads to consider 12 different adaptation problems when we alternatively take each domain as the source or the target dataset. In these experiments we use the same splits as the ones considered in [HRD$^+$13] since they are freely available from the authors website and follow their experimental setup. The results averaged over 20 runs and for each run 8 labelled source examples (20 if the source is Amazon) and 3 labelled target examples are selected. The data is normalized thanks to the zscore and the dimensionality is reduced to 20 thanks to a simple PCA. The results are presented in Table 3 where we compare the performance of our algorithm to 6 baselines : (i) 1-NN$_{\mathcal{S}}$: a 1-NN using the source examples, (ii) 1-NN$_{\mathcal{T}}$: a 1-NN using the target examples, (iii) LMNN$_{\mathcal{T}}$: a 1-NN on the target examples using the metric learned by LMNN on the source examples, (iv) ITML$_{\mathcal{T}}$: a 1-NN on the target examples using the metric learned by ITML on the source examples, (v) MMDT : a domain adaptation method [HRD$^+$13], (vi) GFK : another domain adaptation approach [GSSG12].

The last two methods need the source sample while in our case we only use a source metric learned from the source instances. We consider 3 possible source metrics for our biased regularization framework : (i) Mahalanobis : Inverse of the variance-covariance matrix computed on the source examples, (ii) LMNN : the metric learned by LMNN on the source examples, (iii) ITML : the metric learned by ITML on the source examples.

These results show that metric hypothesis transfer learning can perform well in a Semi-supervised Domain

	Baselines						Our approach		
Task	1-NN$_{\mathcal{S}}$	1-NN$_{T}$	LMNN$_{\mathcal{T}}$	ITML$_{\mathcal{T}}$	MMDT	GFK	MAHALANOBIS	ITML	LMNN
A → C	35.95 ± 1.30	31.92 ± 3.24	32.42 ± 3.03	32.56 ± 4.17	**39.76 ± 2.25**	37.81 ± 1.85	32.65 ± 3.76	32.93 ± 4.60	34.66 ± 3.66
A → D	33.58 ± 4.37	53.31 ± 4.31	49.96 ± 3.53	44.33 ± 8.18	54.25 ± 4.32	51.54 ± 3.55	54.69 ± 3.96	51.54 ± 4.03	**54.72 ± 5.00**
A → W	33.68 ± 3.60	66.25 ± 3.87	62.62 ± 4.49	58.17 ± 10.63	64.91 ± 5.71	59.36 ± 4.30	67.11 ± 5.11	64.09 ± 5.20	**67.62 ± 5.18**
C → A	37.37 ± 2.95	47.28 ± 4.15	42.97 ± 3.76	45.16 ± 7.60	**51.05 ± 3.38**	46.36 ± 2.94	50.15 ± 4.87	49.89 ± 5.25	50.36 ± 4.67
C → D	31.89 ± 5.77	54.17 ± 4.76	46.02 ± 6.54	48.07 ± 8.98	52.80 ± 4.84	**58.07 ± 3.90**	56.77 ± 4.63	53.78 ± 7.23	57.44 ± 4.48
C → W	28.60 ± 6.13	65.06 ± 6.27	55.79 ± 5.09	59.21 ± 9.71	62.75 ± 5.19	63.26 ± 5.89	64.64 ± 6.44	64.00 ± 6.08	**65.11 ± 5.25**
D → A	33.59 ± 1.77	47.81 ± 3.56	40.57 ± 3.79	45.06 ± 6.78	**50.39 ± 3.40**	40.77 ± 2.55	49.48 ± 4.41	49.11 ± 4.09	49.67 ± 4.00
D → C	31.16 ± 1.19	32.22 ± 2.98	27.96 ± 3.03	29.93 ± 4.84	**35.70 ± 3.25**	30.64 ± 1.98	32.90 ± 3.14	32.99 ± 3.58	33.84 ± 2.99
D → W	**76.92 ± 2.18**	66.19 ± 4.60	65.36 ± 3.82	66.74 ± 7.16	74.43 ± 3.10	74.98 ± 2.89	65.57 ± 4.52	66.38 ± 6.04	69.72 ± 3.78
W → A	32.19 ± 3.04	48.25 ± 3.52	41.69 ± 3.71	45.11 ± 5.72	50.56 ± 3.66	43.26 ± 2.34	50.80 ± 3.63	50.16 ± 4.32	**50.92 ± 4.00**
W → C	27.67 ± 2.58	30.74 ± 3.92	28.60 ± 3.41	28.99 ± 4.31	**34.86 ± 3.62**	29.95 ± 3.05	31.54 ± 3.60	31.40 ± 4.29	32.64 ± 3.52
W → D	64.61 ± 4.30	54.84 ± 5.17	56.89 ± 5.06	57.76 ± 7.03	62.52 ± 4.40	**71.93 ± 4.07**	57.17 ± 6.50	56.85 ± 5.51	61.14 ± 5.78
Mean	38.93 ± 3.26	49.84 ± 4.20	45.90 ± 4.11	46.76 ± 7.09	**52.83 ± 3.93**	50.66 ± 3.28	51.12 ± 4.55	50.26 ± 5.02	52.32 ± 4.36

TABLE 3 – Metric Learning for Semi-Supervised Domain Adaptation. For the sake of readability we design the considered domains by their initials. $\mathcal{S} \to \mathcal{T}$ stands for adaptation from the source domain to the target domain. Each time we consider the mean and standard deviation over 20 runs. For each task, the best result is highlighted with a bold font.

Adaptation setting. Indeed, we obtain accuracies which are competitive with state of the art approaches like MMDT or GFK while using less information. Moreover we also perform better than directly plugging the metrics learned by LMNN and ITML in a 1-nearest neighbour classifier.

If we compare the performances of both ITML and LMNN as metrics used directly in a nearest neighbour classifier one can intuitively expect ITML to be a better source hypothesis than LMNN. However, in practice using the metric learned by LMNN as the source hypothesis yields better results. This suggests that using a learned source model that tends to overfit reasonably the learning source sample can be of potential interest in a transfer learning context. Indeed LMNN does not use a regularization term in its formulation and it is well know that LMNN is prone to overfitting. Since, the parameter β penalizes the source metric w.r.t. its complexity it may limit the impact of the source metric to what is needed for the transfer. Nevertheless, this point deserves further investigation.

6 Conclusion

In this paper we presented a new theoretical analysis for metric hypothesis transfer learning. This framework takes into account a source hypothesis information to help learning by means of a biased regularization. This biased regularization can be interpreted into two ways : (i) when the source metric is an a priori known metric such as the identity matrix, the objective is to infer a new metric that performs better than the source metric, (ii) when the source metric has been learned from another domain, the formulation allows one to transfer the knowledge from the source metric to the new domain. This last interpretation refers to a transfer learning setting where the learner does not have access to source examples and can only make use of the source model in the presence of few labelled data.

Our analysis has shown that this framework is theoretically well founded and that a good source hypothesis can facilitate fast generalization in $\mathcal{O}(1/n)$. Moreover, we have provided a consistency analysis from which we have developed a generalization bound able to consider both the performance and the complexity of the source hypothesis. This has led to the use of weighted source hypothesis to optimize the bound in a theoretically sound way.

As stated in [KO14] in another context, our results stress the importance of choosing good source hypothesis. However, choosing the best source metric from few labelled data is a difficult problem of crucial importance. One perpective could be to consider notions of reverse validations as used in some transfer learning/domain adaptation tasks [BM10, ZFY$^+$10]. Another perspective would be to extend our framework to other settings and other kind of regularizers.

Références

[BBC$^+$10] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine Learning*, 2010.

[BE02] Olivier Bousquet and André Elisseeff. Stability and generalization. *JMLR*, 2002.

[BH15] Aurélien Bellet and Amaury Habrard. Robustness and Generalization for Metric Learning. *Neurocomputing*, 2015.

[BHS12] Aurélien Bellet, Amaury Habrard, and Marc Sebban. Similarity learning for provably accurate sparse linear classification. In *Proc. of ICML*, 2012.

[BHS13] Aurélien Bellet, Amaury Habrard, and Marc Sebban. A survey on metric learning for feature vectors and structured data. *CoRR*, 2013.

[BM10] Lorenzo Bruzzone and Mattia Marconcini. Domain adaptation problems : A DASVM classification technique and a circular validation strategy. *Transaction Pattern Analysis and Machine Intelligence*, 2010.

[BYGP14] Julien Bohné, Yiming Ying, Stéphane Gentric, and Massimiliano Pontil. Large margin local metric learning. In *Proc. of ECCV*, 2014.

[CGY12] Qiong Cao, Zheng-Chu Guo, and Yiming Ying. Generalization bounds for metric and similarity learning. *CoRR*, 2012.

[CYL13] Qiong Cao, Yiming Ying, and Peng Li. Similarity metric learning for face recognition. In *Proc. of ICCV*, 2013.

[DKJ+07] Jason V. Davis, Brian Kulis, Prateek Jain, Suvrit Sra, and Inderjit S. Dhillon. Information-theoretic metric learning. In *Proc. of ICML*, 2007.

[GSSG12] Boqing Gong, Yuan Shi, Fei Sha, and Kristen Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *Proc. of CVPR*, 2012.

[GY14] Zheng-Chu Guo and Yiming Ying. Guaranteed classification via regularized similarity learning. *Neural Computation*, 2014.

[HRD+13] Judy Hoffman, Erik Rodner, Jeff Donahue, Kate Saenko, and Trevor Darrell. Efficient learning of domain-invariant image representations. *CoRR*, 2013.

[JWZ09] Rong Jin, Shijun Wang, and Yang Zhou. Regularized distance metric learning : Theory and algorithm. In *Proc. of NIPS*, 2009.

[KC06] Wolf Kienzle and Kumar Chellapilla. Personalized handwriting recognition via biased regularization. In *Proc. of ICML*, 2006.

[KO13] Ilja Kuzborskij and Francesco Orabona. Stability and hypothesis transfer learning. In *Proc. of ICML*, 2013.

[KO14] Ilja Kuzborskij and Francesco Orabona. Learning by transferring from auxiliary hypotheses. *CoRR*, 2014.

[Kul13] Brian Kulis. Metric learning : A survey. *Foundations and Trends in Machine Learning*, 2013.

[PHMS14] Michaël Perrot, Amaury Habrard, Damien Muselet, and Marc Sebban. Modeling perceptual color differences by local metric learning. In *Proc. of ECCV*, 2014.

[PW10] Shibin Parameswaran and Kilian Q. Weinberger. Large margin multi-task metric learning. In *Proc. of NIPS*, 2010.

[SSBD14] Shai Shalev-Shwartz and Shai Ben-David. *Understanding Machine Learning - From Theory to Algorithms*, chapter Regularization and Stability. Cambridge University Press, 2014.

[WBS05] Kilian Q. Weinberger, John Blitzer, and Lawrence K. Saul. Distance metric learning for large margin nearest neighbor classification. In *Proc. of NIPS*, 2005.

[XNJR02] Eric P. Xing, Andrew Y. Ng, Michael I. Jordan, and Stuart J. Russell. Distance metric learning with application to clustering with side-information. In *Proc. of NIPS*, 2002.

[ZFY+10] ErHeng Zhong, Wei Fan, Qiang Yang, Olivier Verscheure, and Jiangtao Ren. Cross validation framework to choose amongst models and datasets for transfer learning. In *Proc. of ECML/PKDD*, 2010.

[ZMW+09] Zheng-Jun Zha, Tao Mei, Meng Wang, Zengfu Wang, and Xian-Sheng Hua. Robust distance metric learning with auxiliary knowledge. In *Proc. of IJCAI*, 2009.